



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/630,339

07/30/2003

Jeff G. Bone

STOR1110-1

9306

44654 7590 03/31/2008

SPRINKLE IP LAW GROUP  
1301 W. 25TH STREET  
SUITE 408  
AUSTIN, TX 78705

EXAMINER

DWIVEDI, MAHESH H

ART UNIT

PAPER NUMBER

2168

MAIL DATE

DELIVERY MODE

03/31/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/630,339	<b>Applicant(s)</b> BONE ET AL.	
	<b>Examiner</b> MAHESH H. DWIVEDI	<b>Art Unit</b> 2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 01 February 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 37 and 39-105 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 37 and 39-105 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 30 July 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

## DETAILED ACTION

### *Continued Examination Under 37 CFR 1.114*

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/5/2006 has been entered.

### *Remarks*

2. Receipt of Applicant's Amendment, filed on 02/01/2008, is acknowledged. The amendment includes the withdrawal of claims 1-36 and 106-113, the cancellation of claim 38, and the amending of claims 37, 55, 72, and 89.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 37, 39-43, 52-59, 69-76, 86-93, and 103-105 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Heilig et al.** (U.S. PG PUB 2002/0046262) in view of **Britton et al.** (U.S. Patent 6,654,814).

Art Unit: 2168

6. Regarding claim 37, **Heilig** teaches a system comprising:

- A) a network (Paragraph 50);
- B) a plurality of client computers (Paragraph 58);
- C) each client computer comprising: a client processor (Paragraph 58);
- D) a client network interface to connect to and interface with the network (Paragraphs 50 and 58);
- E) a client computer readable medium accessible by the client processor, storing a client program executable by the client processor to: generate a first filesystem request pertaining to a filesystem (Paragraph 103);
- F) receive a first filesystem response pertaining to the filesystem (Paragraph 116);
- G) an intermediary device comprising: an intermediary processor (Paragraph 116);
- H) an intermediary network interface to connect to and interface with the network (Paragraph 116);
- I) an intermediary computer readable medium accessible by the intermediary processor and executable to: provide a client-facing filesystem interface (Paragraph 102);
- J) provide a server-facing filesystem interface (Paragraphs 117-118 and 122);
- K) receive the first filesystem request from a requesting client according to the client-facing filesystem interface (Paragraph 103);
- L) pass the first filesystem request to a server as a proxy request according to the server-facing filesystem interface (Paragraph 107);
- O) receive a server response from the server according to the server facing interface (Paragraphs 120 and 124);
- P) pass the server response to the requesting client as the first filesystem response (Paragraph 124);
- Q) a plurality of servers (Paragraph 31);
- R) each server further comprising: a server processor (Paragraph 31);
- S) a server interface coupled to the server processor to connect to and interface with the network (Paragraphs 117-118 and 122);
- T) a server computer readable medium storing a server program executable by the server processor to: provide an origin filesystem (Paragraph 31);

- U) receive the proxy request from the intermediary device (Paragraph 120);
- V) execute a requested filesystem operation (Paragraph 120);
- W) generate the server response (Paragraphs 120 and 124); and
- X) communicate the server response to the intermediary computer (Paragraph 124).

**Heilig** does not explicitly teach:

- M) wherein passing the first filesystem request as a proxy request comprises applying a set of rules to the first filesystem request to determine if the first filesystem request should be modified;
- N) and if it is determined that the first filesystem request should be modified, modifying the first filesystem request to generate the proxy request.

**Britton**, however, teaches **“wherein passing the first filesystem request as a proxy request comprises applying a set of rules to the first filesystem request to determine if the first filesystem request should be modified”** as “As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or “tailoring” (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as “session specific information.” After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first

request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)” (Column 11, lines 1-32), and **“and if it is determined that the first filesystem request should be modified, modifying the first filesystem request to generate the proxy request”** as “As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or "tailoring" (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)” (Column 11, lines 1-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

**Britton's** would have allowed **Heilig's** to provide a method to improve the custom-tailoring of client-requested data in order to better exploit the resources available to the clients, as noted by **Britton** (Column 3, lines 13-15).

Regarding claim 39, **Heilig** does not explicitly teach a system comprising:  
A) wherein the intermediary program is executable to apply active rules to the first filesystem request.

**Britton**, however, teaches “**wherein the intermediary program is executable to apply active rules to the first filesystem request**” as “As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or "tailoring" (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)” (Column 11, lines 1-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Britton's** would have allowed **Heilig's** to provide a method to improve the custom-tailoring of client-requested data in order to better exploit the resources available to the clients, as noted by **Britton** (Column 3, lines 13-15).

Regarding claims 40, 59, 76, and 93, **Heilig** further teaches a system, an intermediary device, device, and method comprising:

A) modifying the server response to generate the proxy response (Paragraphs 61 and 67).

The examiner notes that **Heilig** teaches “**modifying the server response to generate the proxy response**” as “the proxy server may utilize information included in the client data request to determine whether a rendering, i.e. further processing or rewriting of the data is necessary before transmission to the client” (Paragraph 61).

Regarding claims 41, 56, 73, and 90, **Heilig** further teaches a system, an intermediary device, device, and method comprising:

A) determining whether to further process the filesystem request (Paragraph 115);

B) generating a redirect reply (Paragraphs 133-135); and

C) communicating the redirect reply to the requesting client (Paragraphs 133-135).

The examiner notes that **Heilig** teaches “**determining whether to further process the filesystem request**” as “In case the determining module 422 concludes that the request received from the client 102i does not require any rendering operations...the proxy server 420 may directly transmit the requested data to the client device” (Paragraph 115). The examiner further notes that **Heilig** teaches “**generating a redirect reply**” as “The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133). The examiner further notes that **Heilig** teaches “**communicating the redirect reply to the requesting client**” as “The proxy server 420 then generates a dummy response or link



message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133).

Regarding claim 42, **Heilig** further teaches a system comprising:

- A) wherein the client program at each client is further executable to: receive the redirect response as the first filesystem response (Paragraphs 133-135);
- B) generate a second filesystem request (Paragraphs 133-135); and
- C) communicate the second filesystem request to the origin server (Paragraphs 133-135).

The examiner notes that **Heilig** teaches “**wherein the client program at each client is further executable to: receive the redirect response as the first filesystem response**” as “The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133). The examiner further notes that **Heilig** teaches “**generate a second filesystem request**” as ““The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133). The examiner further notes that **Heilig** teaches “**communicate the second filesystem request to the origin server**” as “The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133).

Regarding claim 43, **Heilig** further teaches a system comprising:

- A) wherein the server program at the origin server is further executable to: receive the second filesystem request (Paragraphs 133-135);
- B) execute a requested operation (Paragraphs 133-135);
- C) generate a second server response (Paragraphs 133-135); and
- D) pass the second server response to the requesting client (Paragraphs 133-135).

The examiner notes that **Heilig** teaches “**wherein the server program at the origin server is further executable to: receive the second filesystem request**” as “The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133). The examiner further notes that **Heilig** teaches “**execute a requested operation**” as ““The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133). The examiner further notes that **Heilig** teaches “**generate a second server response**” as “The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133). The examiner further notes that **Heilig** teaches “**pass the second server response to the requesting client**” as “The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410” (Paragraph 133).

Regarding claims 52, 69, 86, and 103, **Heilig** further teaches a system, an intermediary device, device, and method comprising:

- A) comparing the filesystem request to a programmable rulebase to determine if the filesystem request matches a pattern (Paragraphs 149-150); and
- B) if the filesystem request matches a pattern, executing an action associated with the pattern (Paragraphs 149-150).

The examiner notes that **Heilig** teaches “**comparing the filesystem request to a programmable rulebase to determine if the filesystem request matches a pattern**” as “in the event that the client generates a data request concerning a document exceeding a predetermined size, the user may set a preference to render the data” (Paragraph 150). The examiner further notes that **Heilig** teaches “**if the filesystem request matches a pattern, executing an action associated with the**

**pattern**” as “in the event that the client generates a data request concerning a document exceeding a predetermined size, the user may set a preference to render the data” (Paragraph 150).

Regarding claims 53, 70, 87, and 104, **Heilig** further teaches a system, an intermediary device, device, and method comprising:

A) executing the action out-of-band (Paragraph 156).

The examiner notes that **Heilig** teaches “**executing the action out-of-band**” as “The proxy server may still retrieve at least some of the requested data, for example a part of the requested data including data type information, until a decision on rendering is possible and then stop retrieving the requested data” (Paragraph 156).

Regarding claims 54, 71, 88, and 105, **Heilig** further teaches a system, an intermediary device, device, and method comprising:

A) executing the action in-band (Paragraphs 149-150).

The examiner notes that **Heilig** teaches “**executing the action in-band**” as “in the event that the client generates a data request concerning a document exceeding a predetermined size, the user may set a preference to render the data” (Paragraph 150).

Regarding claim 55, **Heilig** teaches an intermediary device comprising:

- A) a processor (Paragraph 116);
- B) a network interface to connect to and interface with a network (Paragraph 50);
- C) a computer readable medium accessible by the processor and executable to: provide a client-facing filesystem interface (Paragraph 102);
- D) provide a server-facing filesystem interface (Paragraphs 117-118, and 122);
- E) receive a filesystem request pertaining to a filesystem from a requesting client according to the client-facing filesystem interface (Paragraph 103);
- F) pass the filesystem request to a server as a proxy request according to the server-facing filesystem interface (Paragraph 107);

Art Unit: 2168

I) receive a server response from the server according to the server-facing interface (Paragraphs 120 and 124); and

J) pass the server response to the requesting client as a proxy response (Paragraph 124).

The examiner notes that **Heilig** teaches “**a computer readable medium accessible by the processor and executable to: provide a client-facing filesystem interface**” as “The client 102*i* may be connected to the wide area network 401 via I/O interface 408” (Paragraph 102). The examiner further notes that **Heilig** teaches “**providing a server-facing filesystem interface**” as “Communication between the client 102*i* and the processing server 410 may include a bitmap protocol or X Windows protocol” (Paragraph 122). The examiner further notes that **Heilig** teaches “**receive a filesystem request pertaining to a filesystem from a requesting client according to the client-facing filesystem interface**” as “The client 102*i* preferably sends requests to the proxy server 420” (Paragraph 103). The examiner further notes that **Heilig** teaches “**pass the filesystem request to a server as a proxy request according to the server-facing filesystem interface**” as “Preferably this involves sending a request from the proxy server 420 to the data server 440” (Paragraph 102). The examiner further notes that **Heilig** teaches “**receive a server response from the server according to the server-facing interface**” as “It is noted that processing server 410 may also be arranged to transmit the rendered data to the client on a return path including the proxy server” (Paragraph 124). The examiner further notes that **Heilig** teaches “**pass the server response to the requesting client as a proxy response**” as “It is noted that processing server 410 may also be arranged to transmit the rendered data to the client on a return path including the proxy server” (Paragraph 124).

**Heilig** does not explicitly teach:

G) wherein passing the first filesystem request as a proxy request comprises applying a set of rules to the first filesystem request to determine if the first filesystem request should be modified;

H) and if it is determined that the first filesystem request should be modified, modifying the first filesystem request to generate the proxy request.

**Britton**, however, teaches **“wherein passing the first filesystem request as a proxy request comprises applying a set of rules to the first filesystem request to determine if the first filesystem request should be modified”** as “As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or "tailoring" (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)” (Column 11, lines 1-32), and **“and if it is determined that the first filesystem request should be modified, modifying the first filesystem request to generate the proxy request”** as “As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or "tailoring"

(i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)" (Column 11, lines 1-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Britton's** would have allowed **Heilig's** to provide a method to improve the custom-tailoring of client-requested data in order to better exploit the resources available to the clients, as noted by **Britton** (Column 3, lines 13-15).

Regarding claims 57, 74, and 91, **Heilig** further teaches an intermediary device, device, and method comprising:

A) wherein the redirect reply is configured to prompt the requesting client to generate a second filesystem request to the server (Paragraphs 133-135).

The examiner notes that **Heilig** teaches "**wherein the redirect reply is configured to prompt the requesting client to generate a second filesystem**

**request to the server**" as "The proxy server 420 then generates a dummy response or link message 521, e.g., in data retrieval module 421, wherein the link message instructs the client to redirect the data request to the processing server 410" (Paragraph 133).

Regarding claims 58, 75, and 92, **Heilig** does not explicitly teach an intermediary device, device, and method comprising:

A) modifying the filesystem request to generate the proxy request.

**Britton**, however, teaches "**modifying the filesystem request to generate the proxy request**" as "As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or "tailoring" (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)" (Column 11, lines 1-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Britton's** would have allowed **Heilig's** to provide a method to improve the custom-tailoring of client-requested data in order to better exploit the resources available to the clients, as noted by **Britton** (Column 3, lines 13-15).

Regarding claims 72, and 89, **Heilig** teaches a device, and method comprising:

- A) providing a client-facing filesystem interface (Paragraph 102);
- B) providing a server-facing filesystem interface (Paragraphs 117-118, and 122);
- C) receiving a filesystem request pertaining to a filesystem from a requesting client according to the client-facing filesystem interface (Paragraph 103);
- D) passing the filesystem request to a server as a proxy request according to the server-facing filesystem interface (Paragraph 107);
- G) receiving a server response from the server according to the server-facing interface (Paragraphs 120 and 124); and
- H) passing the server response to the requesting client as a proxy response (Paragraph 124).

The examiner notes that **Heilig** teaches “**providing a client-facing filesystem interface**” as “The client 102*i* may be connected to the wide area network 401 via I/O interface 408” (Paragraph 102). The examiner further notes that **Heilig** teaches “**providing a server-facing filesystem interface**” as “Communication between the client 102*i* and the processing server 410 may include a bitmap protocol or X Windows protocol” (Paragraph 122). The examiner further notes that **Heilig** teaches “**receiving a filesystem request pertaining to a filesystem from a requesting client according to the client-facing filesystem interface**” as “The client 102*i* preferably sends requests to the proxy server 420” (Paragraph 103). The examiner further notes that **Heilig** teaches “**passing the filesystem request to a server as a proxy request according to the server-facing filesystem interface**” as “Preferably this involves sending a request from the proxy server 420 to the data server 440” (Paragraph 102). The examiner further notes that **Heilig** teaches “**receiving a server response from**



**the server according to the server-facing interface**” as “It is noted that processing server 410 may also be arranged to transmit the rendered data to the client on a return path including the proxy server” (Paragraph 124). The examiner further notes that **Heilig** teaches “**passing the server response to the requesting client as a proxy response**” as “It is noted that processing server 410 may also be arranged to transmit the rendered data to the client on a return path including the proxy server” (Paragraph 124).

**Heilig** does not explicitly teach:

- E) wherein passing the first filesystem request as a proxy request comprises applying a set of rules to the first filesystem request to determine if the first filesystem request should be modified;
- F) and if it is determined that the first filesystem request should be modified, modifying the first filesystem request to generate the proxy request.

**Britton**, however, teaches “**wherein passing the first filesystem request as a proxy request comprises applying a set of rules to the first filesystem request to determine if the first filesystem request should be modified**” as “As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or “tailoring” (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be

referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously transmitted session specific information remains valid (block 106)" (Column 11, lines 1-32), and **"and if it is determined that the first filesystem request should be modified, modifying the first filesystem request to generate the proxy request"** as "As seen in FIG. 3, when the browser 52 transmits a request, the client-side proxy 54 receives the request and determines if it is the first request for the current session (block 100). If the request is the first request, then it is determined if the client data processing system is capable of and has a preference for performing the content transformation or "tailoring" (i.e. should content tailoring occur at the client data processing system 50 or at another data processing system) (block 102). This information, along with other information about the client data processing system 50 and the session, such as for example, data processing capability, available memory, display type and size, resource availability, connection type, priorities for requested information, connection duration, or the like, is incorporated into the request (block 104). Client preferences and other session information (blocks 102 and 104) may reside at the client data processing system 50 or they may be obtained from a server during device initialization, at user logon or with each session. A user identification, such as a userid, may also be included in the request (block 106). The information added or otherwise contained in the request may collectively be referred to as "session specific information." After incorporating the session specific information in the request, the request is sent to the server-side proxy 64 (block 108). Returning to block 100 of FIG. 3, if the request from the browser 52 is not the first request, then, if the server side stores the previously transmitted session specific information, the only information which would need to be inserted into the request is the user identification and a session identifier to indicate that the previously

transmitted session specific information remains valid (block 106)” (Column 11, lines 1-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Britton’s** would have allowed **Heilig’s** to provide a method to improve the custom-tailoring of client-requested data in order to better exploit the resources available to the clients, as noted by **Britton** (Column 3, lines 13-15).

7. Claims 44-51, 60-68, 77-85, and 94-102 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Heilig et al.** (U.S. PGPUB 2002/0046262) in view of **Britton et al.** (U.S. Patent 6,654,814) as applied to claims 37, 39-43, 52-59, 69-76, 86-93, and 103-105 and further in view of **Kao** (U.S. Patent 5,870,734).

8. Regarding claims 44, 61, 78, and 95, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

- A) defining an import space comprising one or more of the origin filesystems;
- B) defining an export space comprising one or more union filesystems; and
- C) wherein the one or more union filesystems are based on the one or more origin filesystems in the import space.

**Kao**, however, teaches “**defining an import space comprising one or more of the origin filesystems**” as “The virtual node architecture allows the present system to accommodate diverse file systems by permitting each node to designate an individual physical file storage system” (Column 5, lines 8-11) and “Any directory or file in the present file system is represented by a vnode, in accordance with the virtual node architecture” (Column 6, lines 29-31), “**defining an export space comprising one or more union filesystems**” as “The virtual node architecture allows the present system to accommodate diverse file systems by permitting each node to designate an individual physical file storage system” (Column 5, lines 8-11) and “Any directory or file in the present file system is represented by a vnode, in accordance with the virtual node architecture” (Column 6, lines 29-31), and “**wherein the one or more union filesystems are based on the one or more origin filesystems in the import space**” as “The virtual node architecture allows the present system to accommodate diverse file

Art Unit: 2168

systems by permitting each node to designate an individual physical file storage system” (Column 5, lines 8-11) and “Any directory or file in the present file system is represented by a vnode, in accordance with the virtual node architecture” (Column 6, lines 29-31).

The examiner notes that by having a virtual node architecture representation (see “vnodes”), **Kao’s** method must have an origin filesystem (see “individual physical file storage system”) which is virtually represented as those vnodes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 45, 62, 79, and 96, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) wherein further comprising stack organizing the one or more origin filesystems in the import space into a stack.

**Kao**, however, teaches “**wherein further comprising stack organizing the one or more origin filesystems in the import space into a stack**” as “The Z-stack is constructed by linking (Z-links) the vnodes representing a pre-selected set of directories” (Column 6, lines 29-31).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 46, 63, 80, and 97, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) stack organizing the one or more origin filesystems by subsuming files and directories from lower origin filesystems in the stack into similarly named files and directories from higher origin filesystems in the stack.

**Kao**, however, teaches “**stack organizing the one or more origin filesystems by subsuming files and directories from lower origin filesystems in the stack into similarly named files and directories from higher origin filesystems in the stack**” as “The “Z-Beam\_up” operations copies files in a specified directory at a lower level in the Z-stack to a specified directory at a higher level in the Z-stack” (Column 7, lines 20-24).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 47, 64, 81, and 98, **Heilig** further teaches a system, an intermediary device, device, and method comprising:

A) wherein the filesystem request further comprises the requested operation and a file upon which the requested operation is to occur (Paragraph 58).

The examiner notes that **Heilig** teaches “**wherein the filesystem request further comprises the requested operation and a file upon which the requested operation is to occur**” as “a request from a user device 102*i*, where user device 102*i* can be any one of the plurality of user devices 102A to 102F, specifies (i) a suitable address to the location where the content associated with the request is stored, for example, an address in the form of a uniform resource locator (URL)...the types of data that can be processed and displayed to the user device” (Paragraph 58).

Regarding claims 48, 65, 82, and 99, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) passing the proxy request based on the filesystem request to a topmost origin filesystem in the stack that contains the file upon which the requested operation is to occur.

**Kao**, however, teaches “**passing the proxy request based on the filesystem request to a topmost origin filesystem in the stack that contains the file upon**

**which the requested operation is to occur”** as “If the path names traverses a Z-stack, the lookup procedure starts at the top directory vnode in the stack to search for the desired entry” (Column 6, lines 46-49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 49, 66, 83, and 100, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) passing the proxy request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur.

**Kao**, however, teaches “**passing the proxy request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur**” as “IF the user changes to the parent directory of dir\_cp, the current directory is moved to the directory at the top of the Z-stack” (Column 6, lines 62-64).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 50, 67, 84, and 101, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) flagging a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin filesystems in the stack from becoming visible.

**Kao**, however, teaches “**flagging a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin**

**filesystems in the stack from becoming visible**” as “the original paths are blocked between lower level vnodes in the stack and their original parent directories” (Column 7, lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 51, 68, 85, and 102, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) wherein the particular file and particular other files share a common name.

**Kao**, however, teaches “**wherein the particular file and particular other files share a common name**” as “FIG. 2” (Figure 2).

The examiner notes that Figure 2 of **Kao** teaches multiple directories having files with common names (see “awk” and “liby.a”)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kao’s** would have allowed **Heilig’s** and **Britton’s** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

Regarding claims 60, 77, and 94, **Heilig** and **Britton** do not explicitly teach a system, an intermediary device, device, and method comprising:

A) presenting a union filesystem via the client-facing interface.

**Kao**, however, teaches “**presenting a union filesystem via the client-facing interface**” as “A file system uses a virtual node architecture to create a three-dimensional directory” (Abstract) and “file systems are manipulated through an object called a “vfs”, or virtual file system” (Column 3, lines 17-18).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

**Kao's** would have allowed **Heilig's** and **Britton's** to provide a method for recognition of all mounted file systems for clients, as noted by **Kao** (Column 4, lines 18-22).

***Response to Arguments***

9. Applicant's arguments filed 02/01/2008 have been fully considered but they are not persuasive.

Applicants argue on page 24 that **"By contrast, paragraph 103 of Heilig simply states that the "client 102i preferably sends requests to the proxy server". Heilig does not explicitly teaches "generate" or "filesystem" or "generate a filesystem request".** However, the examiner wishes to refer to Paragraph 103 of **Heilig** which states "The client 102i preferably sends requests to the proxy server 420" (Paragraph 103). The examiner further wishes to state that the limitation merely recites **"the client computer readable medium accessible by the client processor to generate a first filesystem request pertaining to a filesystem"**. The examiner further wishes to state that it is clear that **Heilig's** client system generates a file request (see "client 102i preferably sends requests"). Furthermore, the client sends these requests to a proxy server, i.e. a filesystem. Because the limitation is entirely broad, **Heilig's** method of a client sending requests to a proxy clearly teaches the aforementioned filesystem request.

Applicants argue on page 25 that **"Thus, if the rejection is to be maintained in the next office action, it is respectfully requested that the Examiner explain why the differences between Heilig and the claim limitation...Heilig does not seem to be concerned with problems particular to the networked filesystem environment".** However, the examiner wishes to state that **Heilig** deals with proxy servers in between clients and servers, just like the independent claims. Moreover, the examiner is not relying on inherency, but to the contrary, showing that the entirely broad limitations of a client sending filesystem requests to a filesystem is taught by **Heilig's** client sending a request to a proxy.

Applicants argue on page 25 that **"Applicant respectfully submits that reading a claim in light of the specification, to thereby interpret limitations explicitly recited in the claim, is quite different thing from "reading limitations of**



**the speciation into a claim,”...independent claims 37, 55, 72, and 89 explicitly recite the term “filesystem”, which is not found in paragraph 116 of Heilig. Heilig appears to focus on a different type of client-server communication. It seems quite clear from at least paragraph 113 of Heilig that the request for data as described by Heilig is a request for data and not a filesystem request pertaining to a networked filesystem. Heilig provides a solution to accessing and visualizing data stored at a remote host on a computer network. Nothing in Heilig appears to be applicable to solving the problems particular in a networked filesystem environment".**

However, the examiner wishes to refer to MPEP 2111.01 [R-5] “IT IS IMPROPER TO IMPORT CLAIM LIMITATIONS FROM THE SPECIFICATION. “Though understanding the claim language may be aided by explanations contained in the written description, it is important not to import into a claim limitations that are not part of the claim. For example, a particular embodiment appearing in the written description may not be read into a claim when the claim language is broader than the embodiment.” *Superguide Corp. v. DirecTV Enterprises, Inc.*, 358 F.3d 870, 875, 69 USPQ2d 1865, 1868 (Fed. Cir. 2004). See also *Liebel-Flarsheim Co. v. Medrad Inc.*, 358 F.3d 898, 906, 69 USPQ2d 1801, 1807 (Fed. Cir. 2004)(discussing recent cases wherein the court expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment);*E-Pass Techs., Inc. v. 3Com Corp.*, 343 F.3d 1364, 1369, 67 USPQ2d 1947, 1950 (Fed. Cir. 2003) (“Interpretation of descriptive statements in a patent’s written description is a difficult task, as an inherent tension exists as to whether a statement is a clear lexicographic definition or a description of a preferred embodiment. The problem is to interpret claims in view of the specification’ without unnecessarily importing limitations from the specification into the claims.”); *Altiris Inc. v. Symantec Corp.*, 318 F.3d 1363, 1371, 65 USPQ2d 1865, 1869-70 (Fed. Cir. 2003) (Although the specification discussed only a single embodiment, the court held that it was improper to read a specific order of steps into method claims where, as a matter of logic or grammar, the language of the method claims did not impose a specific order on the

performance of the method steps, and the specification did not directly or implicitly require a particular order). See also paragraph IV., below. When an element is claimed using language falling under the scope of 35 U.S.C. 112, 6th paragraph (often broadly referred to as means or step plus function language)\*\*, the specification must be consulted to determine the structure, material, or acts corresponding to the function recited in the claim. In *re* Donaldson, 16 F.3d 1189, 29 USPQ2d 1845 (Fed. Cir. 1994) (see MPEP § 2181- § 2186).". Therefore, because the claim limitations are clearly broad and not mention any of the limitations in specification that appellant want to import into the claims, the examiner's rationale of refusing to consider features which are not claimed is valid.

Moreover, the examiner wishes to state that in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "**networked** filesystem" and "solving problems particular in a networked filesystem environment") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicants argue on page 26 that **"As is known to those skilled in the art, the term "filesystem" refers a high-level abstraction for organizing files and data. Files are arbitrary containers for data. By contrast, Britton describes a client proxy capable of inserting user identity and session identifier in a request for content (data) prior to sending that request to a server proxy. From Britton, it seems quite clear that the request for content thus modified by the client proxy is not a filesystem request. Like Heilig, nothing in Britton appears applicable to solving the problems particular in a network filesystem environment. One reason why data requests such as those described in Heilig and Britton would have had no effect on networked filesystems is that filesystems do not allow arbitrary riggers and associated activities to be programmed outside of the permissions hard coded in the original implementation of the filesystems"**. However, the examiner wishes to state that in response to applicant's argument that the references

fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "**networked** filesystem", "solving problems particular in a networked filesystem environment", and "filesystems do not allow arbitrary riggers and associated activities to be programmed outside of the permissions hard coded in the original implementation of the filesystems") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

### **Conclusion**

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 6,122,629 issued to **Walker et al.** on 19 September 2000. The subject matter disclosed therein is pertinent to that of claims 37, and 39-105 (e.g., methods to optimize and process client requests).

U.S. Patent 6,463,465 issued to **Nieuwejaar** on 08 October 2002. The subject matter disclosed therein is pertinent to that of claims 37, and 39-105 (e.g., methods to optimize and process client requests).

U.S. Patent 6,085,234 issued to **Pitts et al.** on 04 July 2000. The subject matter disclosed therein is pertinent to that of claims 37, and 39-105 (e.g., methods to optimize and process client requests).

U.S. Patent 6,247,139 issued to **Walker et al.** on 12 June 2001. The subject matter disclosed therein is pertinent to that of claims 37, and 39-105 (e.g., methods to optimize and process client requests).

U.S. Patent 6,161,191 issued to **Slaughter et al.** on 12 December 2000. The subject matter disclosed therein is pertinent to that of claims 37, and 39-105 (e.g., methods to optimize and process client requests).

### **Contact Information**

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

Art Unit: 2168

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tim T. Vo/

Supervisory Patent Examiner, Art Unit 2168

Mahesh Dwivedi

Patent Examiner

Art Unit 2168

March 26, 2008

/Mahesh H Dwivedi/

Examiner, Art Unit 2168